

Online Research @ Cardiff

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository: <https://orca.cardiff.ac.uk/id/eprint/138509/>

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Dawson, James M., Davis, Timothy A. ORCID: <https://orcid.org/0000-0003-4932-9379>, Gomez, Edward L. ORCID: <https://orcid.org/0000-0001-5749-1507> and Schock, Justus 2021. A self-supervised, physics-aware, Bayesian neural network architecture for modelling galaxy emission-line kinematics. Monthly Notices of the Royal Astronomical Society 503 (1) , pp. 574-584. 10.1093/mnras/stab427 file

Publishers page: <http://dx.doi.org//10.1093/mnras/stab427>
<<http://dx.doi.org//10.1093/mnras/stab427>>

Please note:

Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies.

See

<http://orca.cf.ac.uk/policies.html> for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.



A self-supervised, physics-aware, Bayesian neural network architecture for modelling galaxy emission-line kinematics

James M. Dawson,¹★ Timothy A. Davis¹,² Edward L. Gomez^{1,2} and Justus Schock³

¹Cardiff University, School of Physics and Astronomy, The Parade, Cardiff CF24 3AA, UK

²Las Cumbres Observatory, Suite 102, 6740 Cortona Dr, Goleta, CA 93117, USA

³RWTH Aachen University, Templergraben 55, D-52062 Aachen, Germany

Accepted 2021 February 10. Received 2021 February 9; in original form 2020 October 19

ABSTRACT

In the upcoming decades, large facilities, such as the SKA, will provide resolved observations of the kinematics of millions of galaxies. In order to assist in the timely exploitation of these vast data sets, we explore the use of a self-supervised, physics-aware neural network capable of Bayesian kinematic modelling of galaxies. We demonstrate the network's ability to model the kinematics of cold gas in galaxies with an emphasis on recovering physical parameters and accompanying modelling errors. The model is able to recover rotation curves, inclinations and disc scale lengths for both CO and H I data which match well with those found in the literature. The model is also able to provide modelling errors over learned parameters, thanks to the application of quasi-Bayesian Monte Carlo dropout. This work shows the promising use of machine learning, and in particular, self-supervised neural networks, in the context of kinematically modelling galaxies. This work represents the first steps in applying such models for kinematic fitting and we propose that variants of our model would seem especially suitable for enabling emission-line science from upcoming surveys with e.g. the SKA, allowing fast exploitation of these large data sets.

Key words: methods: data analysis – techniques: image processing – galaxies: kinematics and dynamics.

1 INTRODUCTION

In studying galaxy evolution, astronomers often use the atomic Hydrogen (H I) 21-cm line to trace the outermost regions of galactic discs (e.g. Warren, Jerjen & Koribalski 2004; Begum, Chengalur & Karachentsev 2005; Sancisi et al. 2008; Heald et al. 2011; Koribalski et al. 2018). This region can mark the continuous boundary between galaxies and their surrounding environments, including the dark matter haloes within which galaxies are thought to reside. The rotation curves of extended H I discs can be used to begin probing the properties of dark matter haloes as well as allow the detailed modelling of galaxies' mass distributions when coupled with ancillary observations (e.g. van Albada et al. 1985; de Blok et al. 2008). In the local Universe, H I discs are useful in determining the gaseous content of a galaxy as well as allowing astronomers to probe kinematic properties ranging from substructures such as bars, warps, counter-rotating discs, and spiral arms (e.g. Józsa et al. 2007; Spekkens & Sellwood 2007; Di Teodoro & Fraternali 2015; Kamphuis et al. 2015). Molecular gas observations (typically of the CO molecule) can provide a complimentary view of these regions at high resolution, revealing the interplay between these gas phases. H I is typically more extended than molecular gas, however, allowing it to trace environmental properties such as extended tidal features and the existence of dwarf companions (Hibbard et al. 2001; Sancisi et al. 2008; Heald et al. 2011; Serra et al. 2013; Bosma 2016; Koribalski et al. 2018).

The evolution of H I gives astronomers insight into the method by which galaxies accrete material from surrounding environments and how the mass of galaxies builds and evolves through star formation. The next generation of H I survey instruments (e.g. the Square Kilometre Array, Dewdney et al. 2009; Australian Square Kilometre Array Pathfinder, Johnston et al. 2007, 2008; the South African MeerKaroo Array Telescope, Jonas & MeerKAT Team 2016; the Chinese Five-hundred metre Aperture Spherical Telescope, Li & Pan 2016) are poised to collect observations spanning a large look-back time, advancing our H I driven science as well as pushing this field of astronomy firmly into the *Big Data* era.

Currently it is estimated that the Square Kilometre Array (SKA) will collect data on the order of hundreds of petabytes per year. Given that amount of data is not only too much to fully exploit by hand but also too large to store, astronomers should be looking to develop real-time models that can perform efficient science on incoming data. In an ideal world, physical information would be extracted from incoming data automatically, leaving the work of unravelling the prevailing science to astronomers. However, with such large data volumes and time-intensive techniques, how are astronomers to begin moving in a direction in which we can fully exploit the data quality promised by the SKA?

In previous work, we sought to begin addressing this challenge via the application of machine learning (ML, Dawson et al. 2019), and in particular neural networks, to extract kinematic properties of cold gas in galaxies. Models and tools exist to do this kind of work already. With the upcoming data releases from surveys such as the Widefield ASKAP L-Band Legacy All-Sky Blind Survey (WALLABY), it comes as no surprise that kinematic modelling tools (e.g.

* E-mail: dawsonj5@cardiff.ac.uk

3D-BAROLO¹, Di Teodoro & Fraternali 2015; 2D-BAT², Oh et al. 2017; FAT³, Kamphuis et al. 2015; KinMS⁴, Davis et al. 2013; Davis, Zabel & Dawson 2020) have been in use and ongoing development for some time. Yet these models typically require several minutes or more to provide a full kinematic model of a single object, and longer if errors are required, which may prove problematic for kinematic analyses at SKA survey speeds.

In the past decade, ML has become a popular solution to many *Big Data* challenges in galaxy evolution studies (e.g. Dieleman, Willett & Dambre 2015; Ackermann et al. 2018; Domínguez Sánchez et al. 2018a, b; Bekki 2019), but remains an underutilized resource among the galaxy kinematics community. Computer vision, which often utilizes ML techniques, has been successfully applied to kinematic characterization (e.g. Stark et al. 2018). Yet, there is a distinct absence of directly exploiting ML (with the notable exception of a few recent works, e.g. Shen & Bekki 2020). Recently our group has made attempts to exploit the use of ML in this field, featuring the use of convolutional auto-encoders to identify disturbed cold gas in galaxies using data from both simulations and observations (see Dawson et al. 2019). We still have a long way to go in fully exploring the application of ML to galaxy kinematic characterization but it appears to be a promising avenue of research and one which we explore further in this work.

While conventional ML models are capable of high empirical accuracy and low testing time (e.g. Breiman 2001; Krizhevsky, Sutskever & Hinton 2012), they are often highlighted for their slow training times (Lim, Loh & Shih 2000) and, in some cases, reluctance to generalize to unseen data sets (Dinh et al. 2017; Kawaguchi, Pack Kaelbling & Bengio 2017). These qualities are unsuitable for survey tasks proposed for the SKA and therefore, we are required to look at alternative methods that incorporate the benefits of ML, without the drawbacks associated with standard ML practice.

Such an approach may exist in the form of self-supervised learning (Liu et al. 2020), whereby models train themselves without the need for an isolated training set. This has huge benefits in that one does not require long training times on a throw-away-data set, essentially eliminating data wastage. As with all ML approaches, self-supervised learning does have its disadvantages including requiring fixed analytical functions to perform training, as well as results which change depending on when one wishes to evaluate test data throughout the model training procedure. Few pilot tests of these networks exist in astronomy (and even fewer utilizing physics-aware capabilities, e.g. Aragon-Calvo 2019) and none exist in the modelling of galaxy kinematics. In this paper, we present the current results from our first attempts at creating a self-supervised neural network with the primary goal of inferring the kinematic properties of gas discs in galaxies and an emphasis on extracting (simplistic) characteristics of their rotation curves.

This paper is divided into three main sections. Section 2 gives an in-depth description of the model architecture used throughout this work, with emphasis lying on the decoder subnet described in Section 2.4. Section 3 presents the results from testing the network using synthetic and real interferometric observations and Section 4 summarizes the main outcomes of the work presented in this paper as well as proposed avenues for future work.

2 THE MODEL

2.1 Input data

A typical interferometric observation returns visibilities in a complex plane from which one can obtain a 3D datacube consisting of 2D spatial flux observations separated into discrete channels which correspond to observed frequency. It is this channelization that allows astronomers to measure the line-of-sight velocities and hence the kinematic properties of galaxies' gas reservoirs. In practice, one can collapse these datacubes further to create 2D maps that reflect the mean properties of the gas in galaxies. A moment zero (integrated intensity) map is simply a summation along a cube's frequency/velocity dimension

$$\text{Moment zero} = \int I_v dv = \sum I_v, \quad (1)$$

and a moment one (velocity) map is an intensity weighted averaging of the line-of-sight velocities

$$\text{Moment one} = \frac{\int (v) I_v dv}{\int I_v dv} = \frac{\sum (v) I_v}{\sum I_v}. \quad (2)$$

Working directly with the datacubes, or in fact the complex visibilities, would be optimal for any fast pipeline kinematic modelling tool. However, we have chosen to work with moment maps in this work as a first step and to avoid the problems associated with channelized inputs as explained further in Section 4. It should be noted that, because of our choice to use moment maps, the models described in this work are also suitable to analyse optical IFU maps, as they will be handled similarly by the model described in this work and have been shown to encode kinematic information which can be extracted using both analytical and ML approaches (e.g. Stark et al. 2018; Hansen et al. 2020). This will be explored further in future work (Dawson et al., in preparation).

It should be noted that in this work, we are not making attempts to mitigate the effects of 'beam smearing' (Blais-Ouellette et al. 1999; Swaters 1999). During the recovery of datacubes from complex visibilities, the raw observational datacubes are convolved with a *restoring beam* which effectively encodes the complex visibility plane coverage and is, in some ways, analogous to resolution. It is this convolution step which gives rise to 'beam smearing', the effects of which are discussed further in Section 3.1.2 along with implications for interpreting the model results discussed in this work. Counteracting 'beam smearing' will need to be tackled in future work to maximize the effectiveness of models of this type.

2.2 Model aim

An auto-encoder (Rumelhart, Hinton & Williams 1986) is a model composed of two subnets, an *encoder* and a *decoder*. In an *undercomplete auto-encoder*, the encoder subnet extracts features and reduces input images to a constrained number of nodes. This so-called *bottleneck* forces the network to embed useful information about the input images into a non-linear manifold from which the decoder subnet reconstructs the input images and is scored against the input image using a loss function.

The aim of the model used in this work is to extract semantically meaningful information from observational data. Typical approaches using a convolutional auto-encoder (CAE, Masci et al. 2011; such as that presented in Dawson et al. 2019) are powerful for extracting arbitrary (hyperparametric) features that define dataset characteristics. During training, a CAE learns to minimize the

¹<https://editeodoro.github.io/BBarolo/>

²<https://github.com/seheonoh/2dbat>

³<https://github.com/PeterKamphuis/FAT>

⁴<https://github.com/TimothyADavis/KinMSpy>

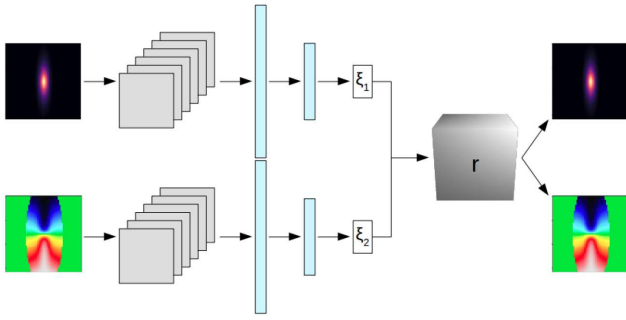


Figure 1. A simplified pictorial representation of the neural network used throughout this work. The model features two convolutional encoder subnets which concatenate learned features before passing them to a decoder subnet. The model receives moment maps as inputs and minimizes the loss between decoder-generated moment map outputs and the inputs throughout training. In the diagram, grey squares indicate convolutional layers, blue rectangles depict linearly connected layers, and the grey cube represents the auxiliary 3D cube containing the coordinate axes passed into the network.

difference between input and output tensors rather than the difference between an output and target label (whether this be a continuous or categorical set of target classes). A CAE works similarly to a powerful non-linear generalization of principle component analysis (PCA, Plaut 2018) whereby it finds a continuous non-linear latent surface on which input data best lies. In this work, however, we would like to extract semantically meaningful parameters of observed systems. In order to achieve this, we have combined a convolutional auto-encoder with a set of analytical, gradient trackable, functions which approximate the functional forms of observed kinematics of galaxies.

The model, known as a *semantic auto-encoder* (SAE, Koldirov, Xiang & Gong 2017), is a modified CAE created using PyTorch⁵ 0.4.1, an open source ML library capable of GPU accelerated tensor computation and automatic differentiation (Paszke et al. 2017). The model has a neural network architecture suited to self-supervised learning, with additional Bayesian capabilities. Fig. 1 shows a simplified pictorial representation of the model architecture.

The encoder subnets extract lower dimensional feature representations from input images (here, the integrated intensity and mean velocity maps as described in Section 2.1) using a combination of convolutional and linearly connected layers; the decoder then reconstructs the input images from the learned feature representations. In a standard convolutional auto-encoder, the decoder would make use of transposed convolution operations, however in this network, the decoder is composed of analytical functions written using native PyTorch. This imposes a constraint on the CAE by forcing the network to generate a semantic encoding of the input images. As highlighted by Aragon-Calvo (2019), the decoder function can take any possible form, no matter how representative of the true underlying functions being modelled. In this way, we can be assured that the encoders are learning semantically meaningful properties of the input images and are no longer tied to traditional training methods, instead allowing the network to train on all available data (including test data) in a self-supervised manner. An SAE becomes physics-aware once the assumption is made that the decoder function can be used to reveal physically meaningful information about the input. In this paper, the physics-awareness of the model refers

Table 1. The SAE encoder subnet architecture used throughout this paper. The first column lists the name of each layer/operation, the second column describes the type of layer/operation, the third column shows the dimensions of each layer’s output tensors (hence the input shape to the next layer). The dimensions follow the PyTorch convention (batch size, number of channels, height, width). The filter column shows the dimensions (height, width) of kernels used to perform the convolution and pooling operations. The convolutional and linearly connected layer groups are separated by a blank row for clarity.

Name	Layer/operation	Dimensions	Filter
Input	–	(64,1,64,64)	–
Conv	2D Convolution	(64,16,64,64)	(3,3)
Pool	2D Max Pooling	(64,16,32,32)	(2,2)
Conv	2D Convolution	(64,32,32,32)	(3,3)
ReLU	ReLU	–	–
Pool	2D Max Pooling	(64,32,16,16)	(2,2)
Conv	2D Convolution	(64,64,16,16)	(3,3)
ReLU	ReLU	–	–
Pool	2D Max Pooling	(64,64,8,8)	(2,2)
Conv	2D Convolution	(64,128,8,8)	(3,3)
ReLU	ReLU	–	–
Pool	2D Max Pooling	(64,128,4,4)	(2,2)
Lc1	Linear	(64,1,1,2048)	–
ReLU	ReLU	–	–
Drop	Dropout ($p = 0.1$)	–	–
Lc2	Linear	(64,1,1,256)	–
Htanh	Hard tanh activation	–	–
Output	–	(64,1,1,2)	–

to our main focus of approximating parametrization for rotation curves, intensity profiles and recovering galaxy inclinations (see Section 2.4).

For a more in-depth background to the use of auto-encoders, we refer the reader to Bourlard & Kamp (1988) and Hinton & Salakhutdinov (2006). For both a concise and thorough introduction to the use of self-supervised, physics aware, neural networks in astronomy, we recommend Aragon-Calvo (2019).

2.3 The encoder subnets

Within the network, the encoders are two convolutional-classifier-like subnets. Each comprises a series of four convolutional and two fully connected layers, interspersed with pooling layers and activation functions. The encoders are used to extract and dimensionally reduce features from input images. The two subnets independently receive a moment zero map (a 2D intensity profile, normalized in the range 0–1) and a moment one map (a 2D velocity profile, normalized into the range –1–1), respectively. Throughout this work, we ensure that the input maps have size of 64×64 pixels. All input maps whose sizes are larger or smaller, like those discussed in Sections 3.2 and 3.3, are subsequently up/down-sampled to a size of 64×64 using PyTorch’s `torch.nn.Upsample` class, in *bilinear* mode. Each moment map carries valuable information for the decoder functions as described in Section 2.4. With this in mind, the output of the encoders are two vectors which are concatenated before passing to the decoder subnet. For an in-depth look at the encoder subnet structure see Table 1.

The encoders learn the following properties: *subnet 1*: observed galaxy inclination (i) and free parameters of the intensity profile which make up ξ_1 in Fig. 1; *subnet 2*: the parameters of the velocity profile of the galaxy which make up ξ_2 in Fig. 1.

⁵<http://pytorch.org/>

2.4 The decoder subnet

Here, we detail the functions required for reconstructing the moment zero and moment one input maps from the concatenated feature representations ξ_1 and ξ_2 as shown in Fig. 1. In recovering the moment maps, we are primarily interested in modelling two profiles. Firstly, the intensity

$$I(r) = I_0 \exp\left(-\frac{r_{x,y}}{r_{\text{scale}}}\right) \exp\left(-\frac{z}{r_{z\text{-scale}}}\right), \quad (3)$$

where I_0 is the intensity normalization factor (set to 1 throughout, due to the global normalization described above), $r_{x,y}$ is the radius in the xy plane, in arcseconds, r_{scale} is the intensity scale length in the xy plane, z is the position in the z axis, and $r_{z\text{-scale}}$ is the intensity scale length in the z axis set to a value of 1 spaxel throughout this work, to emulate a thin disc. Intensity values are determined by combining the integrals of equation (3) across each spaxel in the xy and z planes.

Secondly, the rotational velocity

$$V(r) = \frac{2V_{\text{max}}}{\pi} \arctan\left(-\frac{r}{r_{\text{turn}}}\right), \quad (4)$$

where V_{max} is the asymptotic line of sight velocity, r is the radius in arcseconds, and r_{turn} is the velocity profile scale length.

Here, our choice of exponential intensity profile and arctan velocity profile are entirely arbitrary (i.e. not driven by any physical theory), but are choices motivated by some of the simplest forms that can approximately fit the typical discs and rotation curves found in the Universe. Clearly objects that do not follow these functional forms will not be appropriately fit by this network and we discuss this further in Section 3.5. However, it should be noted that this analytical-style decoder implementation would be equally valid for other functional forms. For example, one could choose to fit bulge-disc models with such an architecture, or include the influence of central point masses or the effects of dark matter haloes. These more realistic networks will be explored in future works.

An auxiliary 3D tensor of radii (labelled r in Fig. 1) is passed into the network, cloned, and evaluated using equations (3) and (4). The 2D moment maps are then created using equations (1) and (2). The velocity profile is later converted into line-of-sight velocity map via an inclination projection and velocity weighting based on the pixel angles about the line-of-sight axis.

2.5 Model training procedure

The network is trained with minimal optimization of hyperparameters in order to demonstrate the simple nature of this architecture. At all times the network utilizes a PyTorch's `MSELoss` function which computes the mean squared error

$$\mathcal{L} = \frac{1}{N} \sum_{i=0}^N (f(x_i) - y_i)^2, \quad (5)$$

between the model outputs, y_i , and inputs, x_i , for every forward pass of a batch of size N . In this case, this is the squared difference between the moment zero and moment one inputs and decoder generated outputs. It is worth noting here that all synthetically generated moment maps have the same position angle and consequently any observational data used for training and testing have been de-rotated using published position angle measurements. We do this as position angle is a non-physical parameter which we can easily account for in pre-processing (with e.g. the `fit_kinematic_pa` routine of Krajnović et al. 2006).

We use an adaptive Adam learning rate optimiser (Kingma & Ba 2014) which begins with a value of 10^{-4} and reduces via multiplication of 0.975 every two epochs. We find that the model converges well after 300 epochs for all training runs presented in this paper.

Where synthetic training data is used, the network receives batches of 64 input moment map pairs. Initial tests showed the network to be largely unaffected by batch size and so 64 is arbitrarily chosen to increase training speed.

The models and PYTHON training scripts used for the work presented in this paper are publicly available on GitHub.⁶

2.6 Model testing procedure

Testing the network can be done in three distinct ways, depending on the situation at hand. In order to test data, one can choose whether to train the network on the test data alone (we call this testing procedure *solo* testing), to train on the test data alongside other examples (we call this testing procedure *combined*), or to use the network in full test mode having only trained on examples not including those data that we wish to test (called *blind* testing).

One can imagine the case where sufficient training data has been passed through the network in a survey, such that in order to return rapid kinematic modelling of new observations, one simply passes the new observations through the network with no prior exposure to the training procedure. This *blind* testing has the advantage of rapid testing speed but at the potential cost of lowered predictive accuracy, in an epistemic uncertainty dominated regime. One can also imagine the case whereby initial survey data has been collected and some sample of the data set the network used to train is also in need of testing. As the network has seen these data during the training procedure, *combined* testing has the advantage of potentially higher accuracy at the expense of time needed to train the model. It should come as no surprise that the ideal testing scenario for this network is *combined*, with a sufficiently large training set in an aleatoric uncertainty dominated regime. However, there are cases (such as at first light of a survey) where the only test data available are that which the network was trained on. It is in this scenario that *solo* testing will occur and although this testing regime lacks the benefits afforded by *combined* testing, it has the potential advantage of predictions not being influenced by anomalous data whose population increases with training set size.

2.7 Monte Carlo dropout

In this section, we summarize the use of *Monte Carlo* dropout (henceforth MC dropout; Gal & Ghahramani 2016) to provide quasi-statistical modelling uncertainties over learned parameters within the model.

In conventional neural network training circumstances, *dropout* may be interpreted as permuting a trained model (Srivastava et al. 2014) via the probabilistic zeroing of weights in linearly connected layers. Traditionally, dropout layers are used throughout training in order to force the network to behave as an ensemble of architectures increased testing accuracy and generalization power. In the case of MC dropout, after training, dropout is reapplied to the network in evaluation mode and inputs are passed through the model many times, effectively sampling a posterior where the model architecture is marginalized out. Gal (2016) first proposed the idea of approximating

⁶<https://github.com/SpaceMeerkat/Coreellia/>

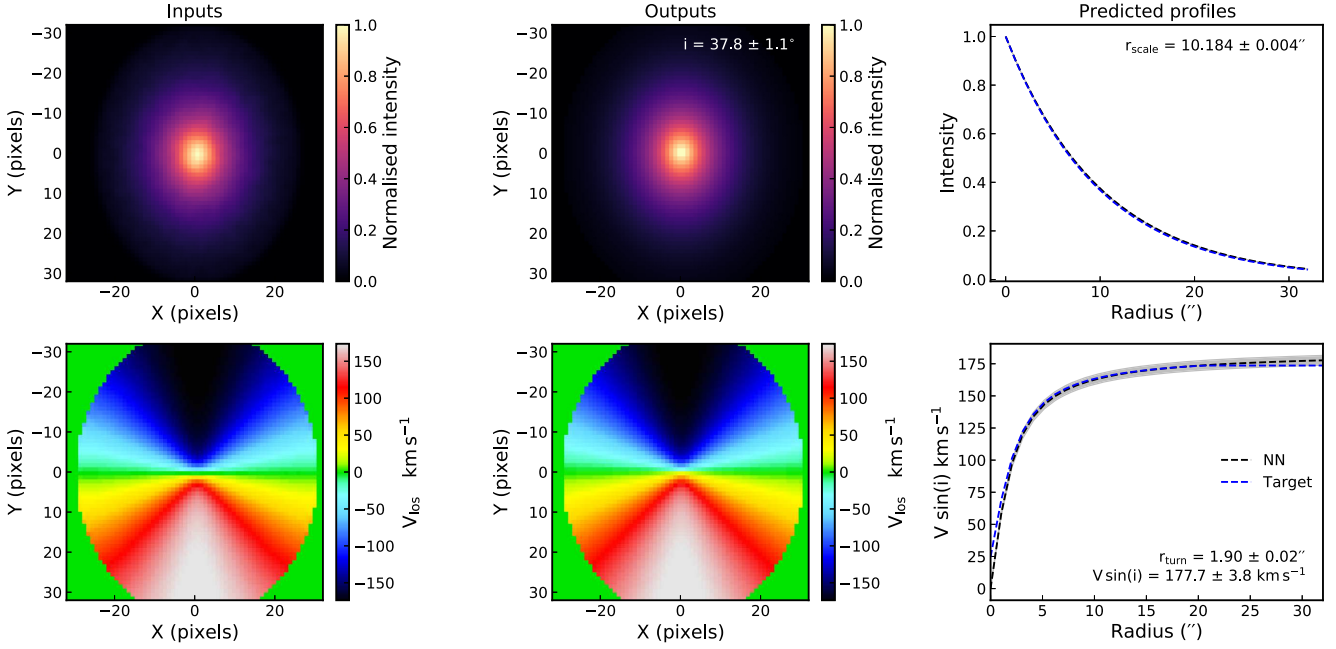


Figure 2. A randomly selected synthesized galaxy, created using *KinMS* and evaluated using the network in *blind* testing mode. The black dashed lines and grey areas show the mean and 1σ modelling uncertainties respectively for profiles predicted by the neural network model. The blue dashed lines show the target profiles which were used to create the input maps. The galaxy was created with the following known parameters: $i = 37.2^\circ$, $r_{\text{scale}} = 10.0''$, $r_{\text{turn}} = 1.6''$, and $V_{\text{max}} \sin(i) = 173.6 \text{ km s}^{-1}$. The network predicted parameters are shown as text in the upper-middle, upper-right, and lower-right subplots.

distributions over parameters learned in neural networks in this way and has since been used in astronomy (e.g. for the probabilistic labelling of galaxy morphologies, Walmsley et al. 2019).

For an input x (comprised of a moment 0 and moment 1 map), training data \mathcal{D} , model weights w , T forward-pass evaluations, and encoder output k , the predicted parameter means and standard deviations are given by equations (6) and (7), respectively.

$$\hat{k} = \frac{1}{T} \sum_i P(k|x, w_i) \quad (6)$$

$$\sigma = \frac{1}{T} \sum_i |k - k_i| \quad (7)$$

For a comprehensive derivation of equations (6) and (7), as well as the implications for using an arbitrary dropout probability, we refer the reader to Walmsley et al. (2019). Examples of the posterior distributions, $p(k|w, \mathcal{D})$, over learned parameters using MC dropout for a randomly selected synthesized galaxy are described further in Section 3.1.1.

It should be noted that, as the network does not use dropout to zero weights in the convolutional layers, σ does not represent a complete error over learned parameters. Instead one should consider σ as a lower limit error over parameters whose use becomes immediately obvious for pipeline flagging purposes or to generate relative errors within a test set. The errors produced through this technique are strictly errors due to the modelling technique, and will underestimate the true error in any parameter, which arises due to both modelling and observational uncertainties.

3 RESULTS AND DISCUSSION

In this section, we present exemplar test results for highly spatially resolved galaxy observations. In each case, we have trained new networks using the procedures described in Section 2.5.

Table 2. Parameter values and ranges for all synthetically generated galaxies using the *KinMS* package. The units for r_{scale} and r_{turn} are absent due to both quantities being fractions of the input map size. The position angle of each galaxy is fixed at 0 as it is not a physically meaningful parameter. Throughout model training, parameters are drawn uniformly in the ranges listed.

Parameter	Size/range	Units
Position angle	0	deg
Inclination	10–90	deg
r_{scale}	0.1–0.35	–
r_{turn}	0.01–0.8	–
$V_{\text{max}} \sin(i)$	50–500	km s^{-1}

3.1 Synthesized examples

3.1.1 Input–output

In order to explore the limitations of the network, we tested the model using synthetic galaxies generated using the PYTHON based kinematic simulator *KinMS*⁷ (KINematic Molecular Simulation, Davis et al. 2013, 2020). Fig. 2 shows the inputs and outputs as well as both known and predicted profiles for a galaxy generated using the same analytical functions described in Section 2.4 with inclination, maximum velocity, and scale lengths drawn randomly in the ranges shown in Table 2, and a fixed beam size of two resolution elements. It is clear that the model is able to recover the galaxy’s rotation curve (and other parameters) well in *blind* testing mode, whereby the model has not yet trained on the test data.

⁷<https://github.com/TimothyADavis/KinMSpy>

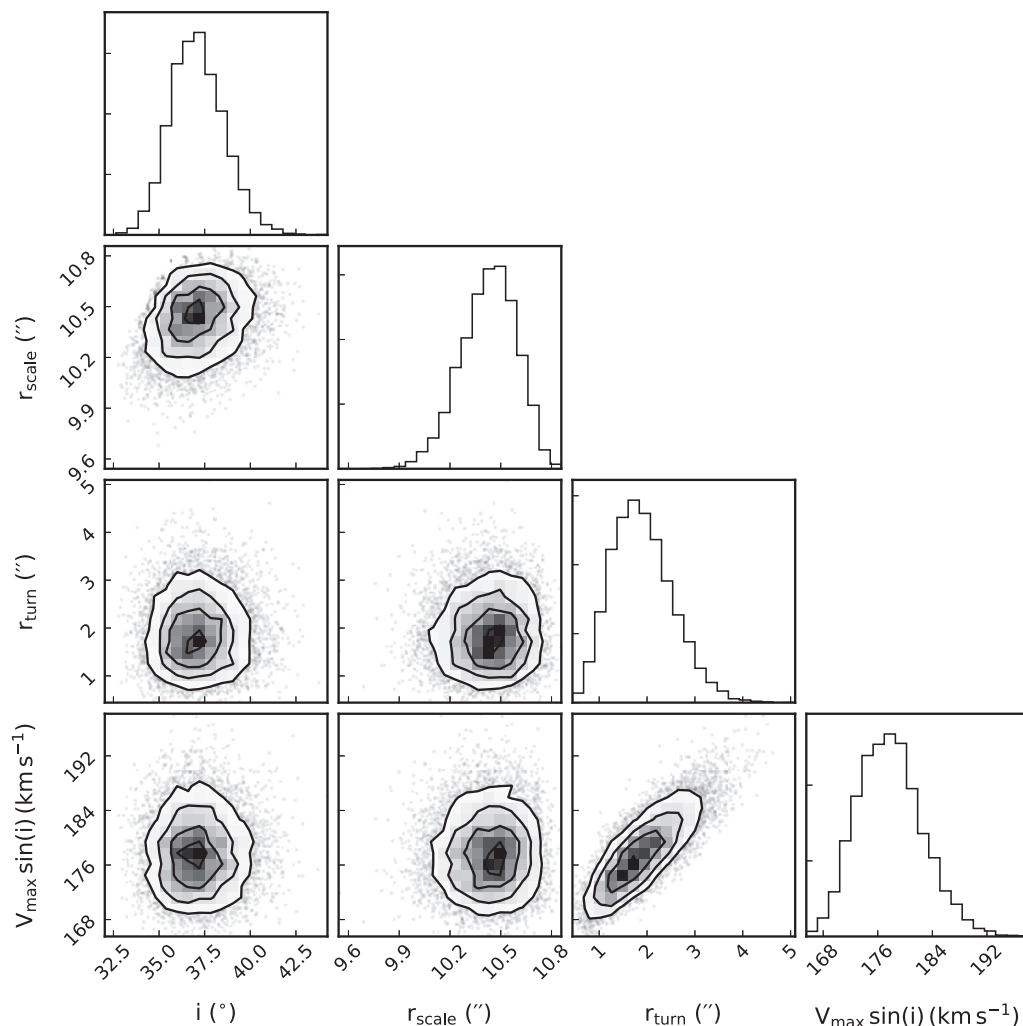


Figure 3. Corner plot showing the level of covariance between learned parameters for one randomly generated, synthesized galaxy (discussed further in Section 3.1). The accompanying histograms represent quasi-probabilistic distributions thanks to the use of *Monte Carlo* dropout. This galaxy was passed through the network in test mode 10 000 times in order to build the distributions. We observe well constrained learned parameters with Gaussian like profiles, allowing for quasi-probabilistic modelling errors for the parameters. The only strong covariance observed is that between the maximum line-of-sight velocity and the velocity profile scale length, which is entirely expected and present in traditional kinematic analyses.

The quasi-probabilistic distributions for each learned parameter for this galaxy are shown in Fig. 3, highlighting the Gaussian-like nature of the learned parameter distributions as well as an expected covariance between r_{turn} and $V_{\text{max}} \sin(i)$.

As seen in Fig. 4, the model is able to recover the desired physical parameters of synthesized galaxies well, heuristically. For the 1739 test galaxies shown in Fig. 4, we measure the average deviation of parameters: i , r_{scale} , r_{turn} , and $V_{\text{max}} \sin(i)$, from the 1:1 line as $\sigma_i = 1.1^\circ$, $\sigma_{r_{\text{scale}}} = 0.004''$, $\sigma_{V_{\text{max}} \sin(i)} = 3.8 \text{ km s}^{-1}$, and $\sigma_{r_{\text{turn}}} = 0.02''$, respectively.

It is clear from Fig. 4 that the error estimates do not represent the total errors over the parameters and only encode the modelling error. This makes the presented errors strictly lower limit estimates, and mostly useful for comparing reliability within the data set, rather than external use. This can be seen by the fact that on average only $\sim 35\%$ of the data points in Fig. 4 have errorbars which overlap with the 1:1 true-versus-predicted line. For the presented data set, these errors likely underestimate the total error by a factor of ~ 2.5 . Including errors in the observations themselves will help to narrow this gap and will be explored further in future work.

3.1.2 The effect of resolution

One expects $r_{\text{scale, pred}}$ to artificially increase with beam size for a fixed r_{scale} . However, r_{scale} is not known for observations of galaxies whose values r_{scale} fall below some fraction of the beamsize. We see this effect happening as shown in Fig. A1 in a non-complex manner. Therefore, we recommend enforcing flagging based on inclination which appears to be strongly linked with those galaxies whose r_{scale} is under predicted (along the minor axis). In the edge-on galaxy case, the minor axis is no longer well resolved resulting in a poor recovery of the intensity profile. However, this is a well-known issue in moment based kinematic modelling, in which the intensity profiles and kinematics can never be fully derived in edge-on galaxies due to line-of-sight effects.

As we have included no method for mitigating the changes induced by varying beam size, it comes as no surprise that the network will behave differently given a sufficiently large ratio of beam size to galaxy extent. Given that we do not have a mechanism for dealing with ‘beam smearing’ in the current network architecture, we expect

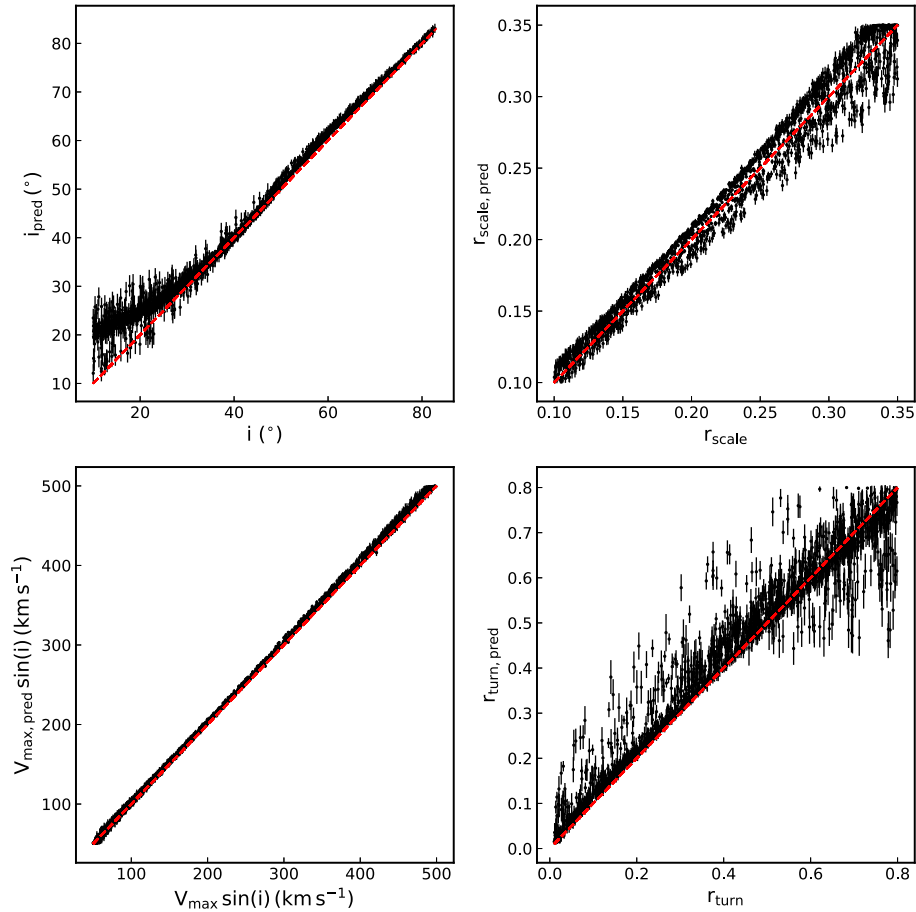


Figure 4. True versus predicted plots for each learnable parameter in the network. Black markers and error bars pertain to the tested galaxies and the red dashed line indicates the 1:1 line on which perfect predictions should ideally lie. This model was trained using purely synthetic data with a restoring beamsize of two resolution elements and only including well resolved examples as discussed in Section 3.1. Those galaxies whose projected r_{turn} fell below 1.5 times the restoring beamsize were removed in order to mimic the automated flagging of poorly resolved galaxies at high inclination in a survey. Of the 2000 synthesized galaxies tested, 261 (13%) were removed using this cut.

to see its influence, lowering the apparent line-of-sight velocities close to the centre of galaxies where the iso-velocity contours are closest together. For minimizing the effects of varying beam size, we recommend convolving the 3D spatial cube r (see Fig. 1), evaluated using equation (3), with the restoring beam before creating the output maps. The advantage of this approach being that the restoring beam is often included in data-product header units, and so should be readily available for creating kernels with which to perform the aforementioned convolution. We consider this approach as beyond the scope of the work presented in this paper, but will be included in future work focusing on retrieving the properties of marginally resolved galaxies.

3.1.3 Fill factor

In previous work, we showed that the fill factor (i.e. the number of zeroed pixels) in a velocity map’s field of view, impacts the behaviour of NN models which take them as inputs (Dawson et al. 2019). With the NN model presented in this work, we have seen little evidence that this has an effect on the galaxies’ predicted parameters. We attribute this behaviour to the nature of the training procedure, whereby in *combined* and *solo* testing, the network does not rely solely upon inference of unseen data.

3.2 H I examples

The primary goal of developing a network like that presented in this paper is to demonstrate the applicability of ML to SKA science. As such, in this section we show the network performs well with H I observational data. In order to do this we present two example test galaxies, NGC 2403 and NGC 3198, observed using the Very Large Array (VLA) as part of The H I Nearby Galaxy Survey (THINGS; Walter et al. 2008), and showing a diversity of rotation curve shapes. These galaxies are 2 of 17 THINGS galaxies used for *mixed* training and testing using the network and chosen heuristically for the appearance of their well defined rotating H I discs. The names and publications for the galaxies used in this sample are shown in Table A1.

Fig. 5 shows the derived intensity profile and rotation curve for NGC 2403. We include the rotation curve modelled using BBarolo (Di Teodoro & Fraternali 2015) on the datacube (Di Teodoro & Lelli, private communication). In comparison, we see that the neural network’s predicted rotation curve matches closely and so we are convinced that the network is able to recover physical information well. Although the galaxy’s intensity profile does not strictly exhibit an exponential form, this has little impact in the recovery of the rotation curve which is the networks primary objective.

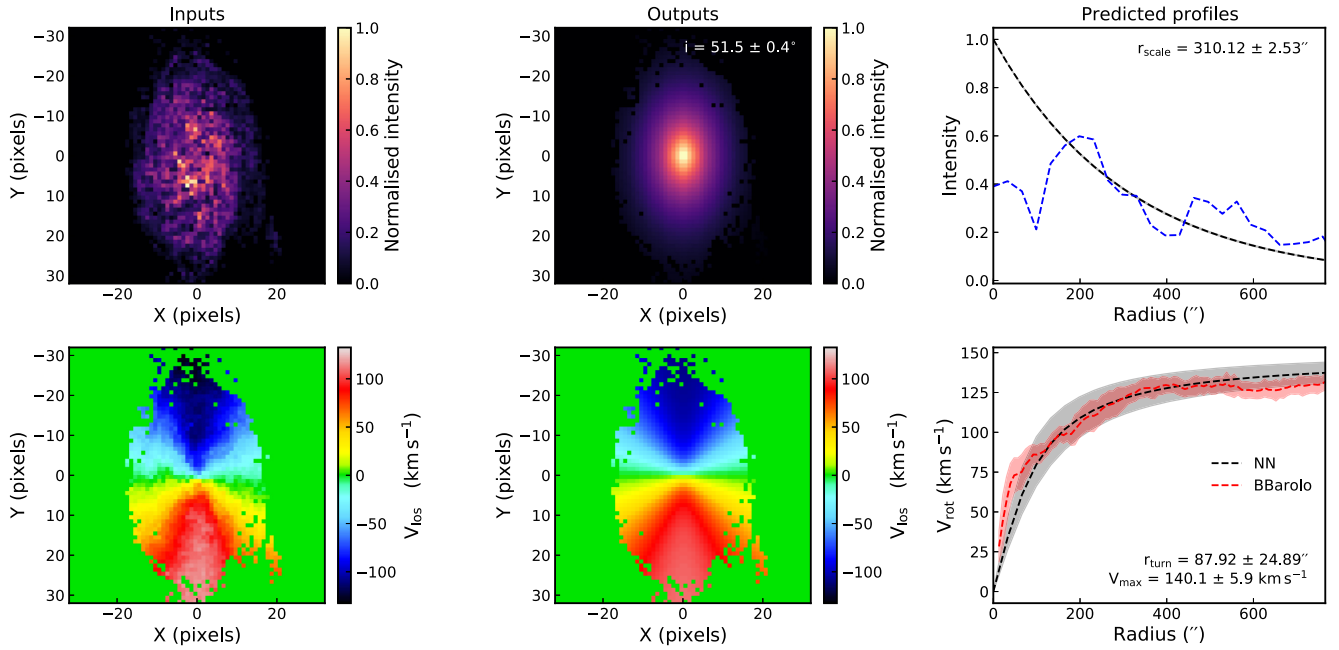


Figure 5. An example galaxy, NGC 2403, observed in H I and evaluated using the network in *combined* testing mode. Maps in the left-hand and middle columns share x - and y -axis sizes of 64×64 pixels. In this way, we are directly observing the input and output maps of the model. The right-hand column has undergone an x -axis rescaling to match observational scales found in the literature. The black dashed lines and grey areas show the mean and 1σ modelling errors, respectively, for profiles predicted by the neural network. The blue dashed line shows a major axis cut of the input intensity map. The red dashed line and filled area show the best fit and associated errors modelled using BBarolo on the datacube. In order to make a direct comparison between the network's and BBarolo's derived rotation curves, the network's velocity profile has been corrected for by the predicted inclination term. The network predicted parameters are shown as text in the upper-middle, upper right-hand, and lower right-hand subplots. We see that this galaxy has a velocity profile which can be roughly approximated by an arctan function meaning the kinematic parameters are well recovered by the model.

Fig. 6 shows the derived intensity profile and rotation curve for NGC 3198. This galaxy exhibits a mild warp and a flat rotation curve (Gentile et al. 2013) with a slight rise at $\sim 200''$. Warped H I discs are not uncommon in the outer regions of galaxies. At present, our network architecture is not set up to model these (however, one could easily extend the model in order to do so). Again, we include the rotation curve modelled using BBarolo on the datacube (Di Teodoro & Lelli, private communication) in Fig. 6. Crucially, although this warping behaviour is not included in our model, in this case, the network still returns reasonable parameter estimations, showing that it could still be usable for parameter estimations across a broadly diverse population of galaxies.

3.3 CO examples

In order to demonstrate the flexibility of this network architecture, we trained a model to recover the kinematic properties of galaxies observed in the CO line using the Atacama Large Millimeter/submillimeter Array (ALMA). Our samples are drawn from the mm-Wave Interferometric Survey of Dark Object Masses (WISDOM) project (see Table A2 for more information) and have high spatial resolution. Due to the nature of these objects being targeted for their evidence of black hole influence on the gas kinematics, we expect to see small values of a_V for the sample. As seen in Fig. 7, this effect is clearly visible, highlighting the predictable behavioural nature of the network. It is also clear in Fig. 7, that NGC 1387 (FCC184, Zabel et al. 2020; Boyce et al., in preparation), an exemplar galaxy from the WISDOM sample, exhibits an exponential intensity profile which the network can easily recover.

Such an example demonstrates the *transferable* nature of this network architecture and training style but without the difficulties often associated with traditional transfer learning tasks. This means that such architectures and training styles can be applied to a multitude of different data sets with the possibility of architectural modifications suiting other types of data outside of interferometry and even astronomy.

3.4 Testing speed

The network can retrieve a mean field approximation for all learnable parameters, of a single galaxy observation, in 0.0025 s on a single Intel(R) Core(TM) i7-6700 CPU core. This time-scales linearly with the number of MC dropout samples one wishes to collect (i.e. for a set of 1000 MC dropout samples, a typical test on an individual galaxy would take 2.5 s) to generate pseudo-probabilistic distributions. However, as the batch throughput size is limited only by the available device memory, it is possible to retrieve values for learnable parameters, and hence MC dropout samples, in the same time frames as listed above for multiple observations. This means that one could potentially return hundreds to thousands of parametrization and associated pseudo-errors in a matter of seconds.

3.5 Caveats

There are a few caveats pertaining to the use of the model described in this work. These caveats may impact the way in which users handle the network and the confidence levels associated with parameter estimations.

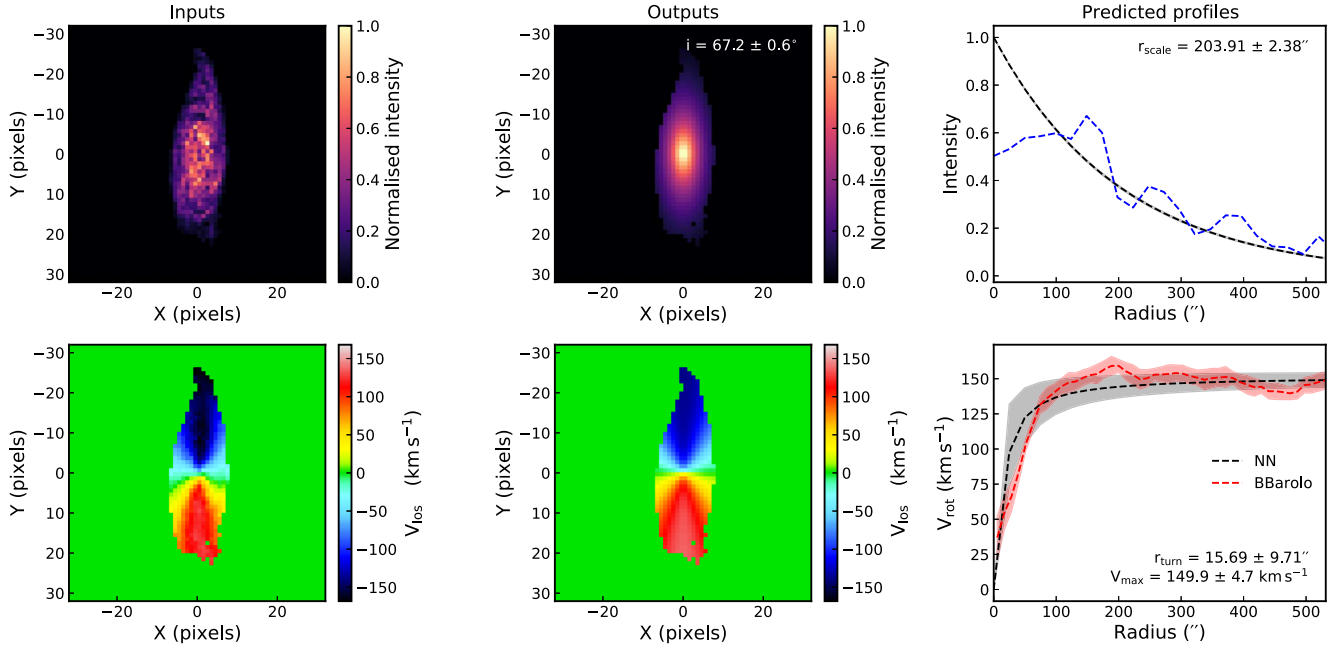


Figure 6. An example galaxy, NGC 3198, observed in H I and evaluated using the network in *combined* testing mode. Maps in the left-hand and middle columns share x - and y -axis sizes of 64×64 pixels. In this way, we are directly observing the input and output maps of the model. The right-hand column has undergone an x -axis rescaling to match observational scales found in the literature. The black dashed lines and grey areas show the mean and 1σ modelling errors, respectively, for profiles predicted by the neural network. The blue dashed line shows a major axis cut of the input intensity map. The red dashed line and filled area show the best fit and associated errors modelled using BBarolo on the datacube. In order to make a direct comparison between the network's and BBarolo's derived rotation curves, the network's velocity profile has been corrected for by the predicted inclination term. The network predicted parameters are shown as text in the upper-middle, upper right-hand, and lower right-hand subplots. We see that this galaxy has a velocity profile which can be roughly approximated by an arctan function meaning the kinematic parameters are well recovered by the model.

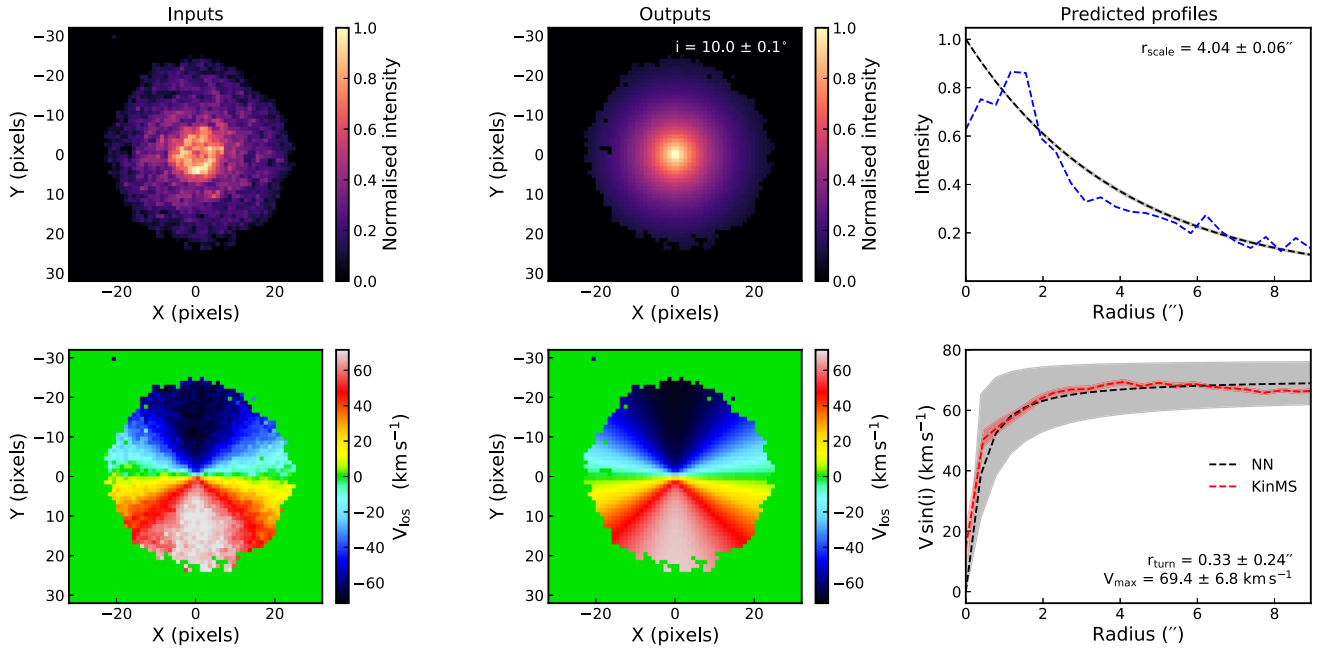


Figure 7. An example WISDOM galaxy, NGC 1387, observed in CO and evaluated using the network in *combined* testing mode. The left-hand and middle columns share x - and y -axis sizes of 64×64 pixels. In this way, we are directly observing the input and output maps of the model. The right-hand column has undergone an x -axis rescaling to match observational scales found in the literature. The black dashed lines and grey areas show the mean and standard deviation, respectively, for profiles predicted by the neural network model. The blue dashed line shows a major axis cut of the input intensity map. The red dashed line shows the KinMS reconstructed rotation curve. The network predicted parameters are shown as text in the upper-middle, upper right-hand, and lower right-hand subplots. We easily see that this galaxy has an intensity profile and velocity profile which can be roughly approximated by an exponential and an arctan function, respectively, meaning the kinematic parameters are well recovered by the model.

A key factor in recovering sensible parametrization using the network is the choice of decoder functions (see Section 2.4). In this work, we have used simple, general, functions in the form of an exponential (see equation 3), and an arctan (see equation 4). However, should one wish to model specific emission line components of galaxies, it would be prudent to adopt more tailored functional forms. For example, it has been shown that H I discs can display depressions in their intensities in their central regions, typically filled by molecular gas (Wong & Blitz 2002), for which a truncated Gaussian intensity profile (Martinsson et al. 2013) would be more appropriate when reconstructing the intensity maps. Additionally, when modelling the very outer regions of H I discs, one might consider adopting a more complex multiparameter function capable of encoding the *sharpness* of the turnover at r_{turn} and the behaviour of the curve after this point (e.g. Rix et al. 1997), or even declining velocities in the central regions (Lelli, McGaugh & Schombert 2016). A declining rotation curve would be challenging for the current model to fit (and impossible to fully retrieve). However, due to the nature of the loss function chosen in this work (see equation 5), the network will prioritize fitting to the higher velocity regions of galaxies.

As described in Section 3.1.2, the resolution of input images impacts the ability of the network to correctly predict a_{scale} , particularly in the high inclination regime. This places constraints on the user's confidence in parameter estimations when working in both the large-beam and high inclination cases combined. Additionally, we can see in Fig. 4 that the network struggles to accurately recover inclinations at the very low inclination range. This is a predictable effect caused by the loss of line-of-sight velocity information for face on discs but again, in the case of survey pipelines, these low inclined galaxies will require additional flagging. In both the aforementioned caveat cases, it is worth noting that traditional kinematic modelling methods also struggle to accurately estimate parameters, in particular, when working with moment maps. Extensions of the network's framework presented here to kinematically model datacubes may alleviate these issues and will be explored in future work.

4 CONCLUSIONS

We have demonstrated the performance of a neural network model architecture which can be used to recover rotation curves of galaxies from their kinematics. The model was tested on synthetically generated galaxies as well as observations using both H I and CO emission lines.

Testing on synthetically generated galaxies has highlighted the powerful performance of the network as well as areas where the network's performance is sub optimal. For the latter areas, we have discussed solutions including: an additional convolution with the restoring beam to counteract the effects of 'beam smearing', and flagging high inclination data in a large beam and high inclination regime.

Testing observational H I data from THINGS has shown that this style of network is well suited to work with data like that expected from the SKA in the near future. We have shown that the network is capable of estimating velocity curves for discs exhibiting a variety of profiles. In order to do this, we have directly compared the rotation curves estimated by the network to those modelled directly from the cubes using kinematic modelling tools. The network is able to perform adequate recovery of parameters even in cases where it would not be possible to reproduce the true rotation curves. These promising results give us confidence that adopting more flexible decoder functions will extend the applicability of the model for more specific use cases should one wish to model H I discs exclusively.

Testing observational CO data from the WISDOM project has shown that the network is suitable for a range of emission line observations. Unlike traditional ML models, the network architecture and training styles outlined in this work prevent the need for *transfer learning* which is often time consuming and fraught with ungainly challenges associated with systematic properties of training sets. We have shown that the model outlined in this work can recover rotation curves which heuristically match rotation curves extracted from ALMA observations using more time-consuming approaches.

As previously stated, improvements to the model architecture in this work include but are not limited to: adapting the model to use more complex intensity and velocity profiles in the decoder subnet, automatically accounting for large beam effects such as beam smearing and information loss either via systematic offsets in model predictions or via the incorporation of an extra convolutional layer in the decoder subnet, and reintroducing a position angle estimation step. An idealized improvement on the model would be to work directly with interferometric datacubes themselves, or even visibilities, without the need to generate moment maps prior to training and testing. However, we have found that the discretized nature of channels in interferometric datacubes presents a non-gradient-trackable step in the decoder's reconstruction of datacube inputs. This discontinuity in the gradient tree prevents back propagation via gradient descent and consequently halts model training. We propose adapting this self-supervised approach to work with datacubes as a lucrative avenue of research for challenging current kinematic modelling tools in preparation for the SKA and other upcoming large facilities.

ACKNOWLEDGEMENTS

This paper has received funding by the Science and Technology Facilities Council as part of the Cardiff, Swansea & Bristol Centre for Doctoral Training.

We gratefully acknowledge the support of Nvidia Corporation with the donation of a Titan Xp GPU used for this research. We also thank the anonymous reviewer whose comments and suggestions helped improve this manuscript.

JMD wishes to gratefully acknowledge the help of Dr Federico Lelli for providing rotation curves for THINGS sample galaxies and valuable insight which have both contributed towards improving this paper.

TAD acknowledges support from the UK Science and Technology Facilities Council through grant ST/S00033X/1.

This research made use of ASTROPY,⁸ a community-developed PYTHON package for Astronomy (Astropy Collaboration 2013, 2018), NUMPY⁹, an open source numerical computation library (Harris et al. 2020), and PANDAS¹⁰, a data manipulation software library (Pandas development team 2020).

This paper makes use of data obtained using the Jansky Very Large Array, a component of the National Radio Astronomy Observatory (NRAO). The NRAO is a facility of the National Science Foundation operated under cooperative agreement by Associated Universities, Inc.

This paper makes use of ALMA data. ALMA is a partnership of the ESO (representing its member states), NSF (USA), and NINS (Japan), together with the NRC (Canada), NSC, ASIAA (Taiwan),

⁸<http://www.astropy.org>

⁹<https://numpy.org/>

¹⁰<https://pandas.pydata.org/>

and KASI (Republic of Korea), in cooperation with the Republic of Chile. The Joint ALMA Observatory is operated by the ESO, AUI/NRAO, and NAOJ.

DATA AVAILABILITY

The data and scripts underlying this article are available via GitHub, at <https://github.com/SpaceMeerkat/Corellia>.

REFERENCES

- Ackermann S., Schawinski K., Zhang C., Weigel A. K., Turp M. D., 2018, *MNRAS*, 479, 415
- Aragon-Calvo M. A., 2019, *MNRAS*, 498, 3713
- Astropy Collaboration, 2013, *A&A*, 558, A33
- Astropy Collaboration, 2018, *AJ*, 156, 123
- Begum A., Chengalur J. N., Karachentsev I. D., 2005, *A&A*, 433, L1
- Bekki K., 2019, *MNRAS*, 485, 1924
- Blais-Ouellette S., Carignan C., Amram P., Côté S., 1999, *AJ*, 118, 2123
- Bosma A., 2016, *Proc. IAU Symp. S321, Formation and Evolution of Galaxy Outskirts*. Kluwer, Dordrecht, p. 220
- Bourlard H., Kamp Y., 1988, *Biologi. Cybern.*, 59, 291
- Breiman L., 2001, *Mach. Learn.*, 45, 5
- Davis T. A., Bureau M., Cappellari M., Sarzi M., Blitz L., 2013, *Nature*, 494, 328
- Davis T. A., Bureau M., Onishi K., Cappellari M., Iguchi S., Sarzi M., 2017a, *MNRAS*, 468, 4675
- Davis T. A. et al., 2017b, *MNRAS*, 473, 3818
- Davis T. A., Zabel N., Dawson J. M., 2020, Astrophysics Source Code Library, record ascl: 2006.003
- Dawson J. M., Davis T. A., Gomez E. L., Schock J., Zabel N., Williams T. G., 2019, *MNRAS*, 491, 2506
- de Blok W. J. G., Walter F., Brinks E., Trachternach C., Oh S. H., Kennicutt R. C. J., 2008, *AJ*, 136, 2648
- Dewdney P. E., Hall P. J., Schilizzi R. T., Lazio T. J. L. W., 2009, *IEEE Proc.*, 97, 1482
- Di Teodoro E. M., Fraternali F., 2015, *MNRAS*, 451, 3021
- Dieleman S., Willett K. W., Dambre J., 2015, *MNRAS*, 450, 1441
- Dinh L., Pascanu R., Bengio S., Bengio Y., 2017, PMLR, 70, 1019
- Domínguez Sánchez H., Huertas-Company M., Bernardi M., Tuccillo D., Fischer J. L., 2018a, *MNRAS*, 476, 3661
- Domínguez Sánchez H. et al., 2018b, *MNRAS*, 484, 93
- Gal Y., 2016, PhD thesis, Univ. Cambridge
- Gal Y., Ghahramani Z., 2016, Advances in Neural Information Processing Systems 29p. 1027
- Gentile et al., 2013, *A&A*, 554, A125
- Hansen S., Conselice C. J., Fraser-McKelvie A., Ferreira L., 2020, *Res. Notes Am. Astron. Soc.*, 4, 185
- Harris C. R. et al., 2020, *Nature*, 585, 357
- Heald G. et al., 2011, *A&A*, 526, A118
- Hibbard J. E., van der Hulst J. M., Barnes J. E., Rich R. M., 2001, *AJ*, 122, 2969
- Hinton G. E., Salakhutdinov R. R., 2006, *Science*, 313, 504
- Johnston S. et al., 2007, *PASA*, 24, 174
- Johnston S. et al., 2008, *Exp. Astron.*, 22, 151
- Jonas J., MeerKAT Team, 2016, in Taylor R., Camilo F., Leeuw L., Moodley K., eds, Proc. Sci., MeerKAT Science: On the Pathway to the SKA. SISSA, Trieste, PoS#001
- Józsa G. I. G., Kenn F., Klein U., Oosterloo T. A., 2007, *A&A*, 468, 731
- Kamphuis P., Józsa G. I. G., Oh S.-H., Spekkens K., Urbancic N., Serra P., Koribalski B. S., Dettmar R.-J., 2015, *MNRAS*, 452, 3139
- Kawaguchi K., Pack Kaelbling L., Bengio Y., 2017, preprint (arXiv:1710.05468)
- Kingma D. P., Ba J., 2014, preprint (arXiv:1412.6980)
- Kodirov E., Xiang T., Gong S., 2017, preprint (arXiv:1704.08345)
- Koribalski B. S. et al., 2018, *MNRAS*, 478, 1611
- Krajinović D., Cappellari M., de Zeeuw P. T., Copin Y., 2006, *MNRAS*, 366, 787
- Krizhevsky A., Sutskever I., Hinton G. E., 2012, Advances in Neural Information Processing Systems 25, p. 1097
- Lelli F., McGaugh S. S., Schombert J. M., 2016, *AJ*, 152, 157
- Li D., Pan Z., 2016, *Radio Sci.*, 51, 1060
- Lim T.-S., Loh W.-Y., Shih Y.-S., 2000, *Mach. Learn.*, 40, 203
- Liu X., Zhang F., Hou Z., Wang Z., Mian L., Zhang J., Tang J., 2020, preprint (arXiv:2006.08218)
- Martinsson T. P. K., Verheijen M. A. W., Westfall K. B., Bershadsky M. A., Andersen D. R., Swaters R. A., 2013, *A&A*, 557, A131
- Masci J., Meier U., Cireşan D., Schmidhuber J., 2011, in Honkela T., Duch W., Girolami M., Kaski S., eds, Artificial Neural Networks and Machine Learning – ICANN 2011, Springer, Berlin, Heidelberg, p. 52
- North E. V. et al., 2019, *MNRAS*, 490, 319
- Oh S.-H., Staveley-Smith L., Spekkens K., Kamphuis P., Koribalski B. S., 2017, *MNRAS*, 473, 3256
- Onishi K., Iguchi S., Davis T. A., Bureau M., Cappellari M., Sarzi M., Blitz L., 2017, *MNRAS*, 468, 4663
- Pandas development team T., 2020, pandas-dev/pandas: Pandas. Available at <https://doi.org/10.5281/zenodo.3509134>
- Paszke A. et al., 2017, NIPS: Autodiff Workshop, available at <https://openreview.net/forum?id=BJJsrnfCZ>
- Plaut E., 2018, preprint (arXiv:1804.10253)
- Rix H.-W., Guhathakurta P., Colless M., Ing K., 1997, *MNRAS*, 285, 779
- Rumelhart D. E., Hinton G. E., Williams R. J., 1986, *Nature*, 323, 533
- Sancisi R., Fraternali F., Oosterloo T., van der Hulst T., 2008, *A&A Rev.*, 15, 189
- Serra P. et al., 2013, *MNRAS*, 428, 370
- Shen A. X., Bekki K., 2020, *MNRAS*, 497, 5090
- Smith M. D. et al., 2019, *MNRAS*, 485, 4359
- Spekkens K., Sellwood J. A., 2007, *ApJ*, 664, 204
- Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., 2014, J. Mach. Learn. Res., 15, 1929
- Stark D. V. et al., 2018, *MNRAS*, 480, 2217
- Swaters R. A., 1999, PhD thesis, Rijksuniversiteit Groningen
- van Albada T. S., Bahcall J. N., Begeman K., Sancisi R., 1985, *ApJ*, 295, 305
- Walmsley M. et al., 2019, *MNRAS*, 491, 1554
- Walter F., Brinks E., Blok W., Bigiel F., Kennicutt R., Thornley M., Leroy A., 2008, *AJ*, 136, 2563
- Warren B. E., Jerjen H., Koribalski B. S., 2004, *AJ*, 128, 1152
- Wong T., Blitz L., 2002, *ApJ*, 569, 157
- Zabel N. et al., 2020, *MNRAS*, 496, 2155

APPENDIX A: EXTRA MATERIAL

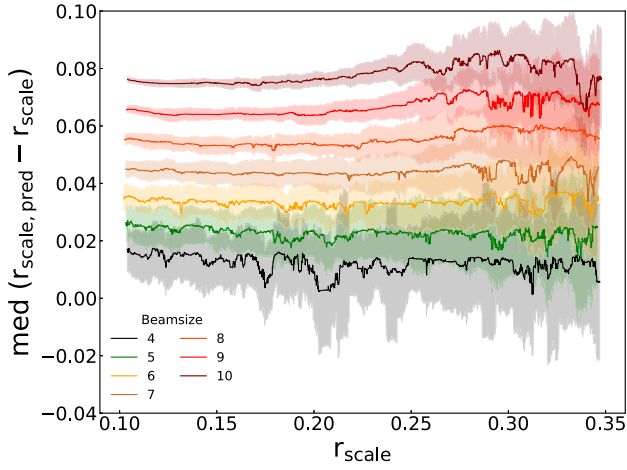


Figure A1. The effects of varying the ratio of beam size to galaxy extent. It is clear to see that an increased beam size results in an artificial lengthening of the intensity profile scale length. It can also be seen that the spread in median offset increases with r_{scale} , which occurs due to information loss as the convolved flux is ‘smeared’ out beyond the field of view. The value of r_{scale} at which this effect begins to take hold is clearly inversely proportional to the beamsizes.

Table A1. Information regarding the THINGS sample galaxies used throughout this work. Columns give the following information: *Object*, the target name as given in THINGS project publications, *Publication* records the relevant publication in which the THINGS targets appear.

Object		Publication
DDO 53	NGC 3621	All from Walter et al. (2008), de Blok et al. (2008)
NGC 925	NGC 4736	
NGC 2403	NGC 4826	
NGC 2841	NGC 5055	
NGC 2903	NGC 5236	
NGC 3184	NGC 6946	
NGC 3198	NGC 7331	
NGC 3351	NGC 7793	
NGC 3521		

Table A2. Information regarding the WISDOM project sample used throughout this work. Table columns give the following information: *Object*, the target name as given in WISDOM project publications, *Observation type* gives the emission line ALMA observed for the target, *Publication* records the relevant publication in which ALMA observations of the targets appear.

Object	Observation type	Publication
NGC 3665	$^{12}\text{CO}(2-1)$	Onishi et al. (2017)
NGC 0383	$^{12}\text{CO}(2-1)$	North et al. (2019)
NGC 0524	$^{12}\text{CO}(2-1)$	Smith et al. (2019)
NGC 1387	$^{12}\text{CO}(2-0)$	Zabel et al. (2020), Boyce et al. (in preparation)
NGC 4429	$^{12}\text{CO}(3-2)$	Davis et al. (2017b)
NGC 4697	$^{12}\text{CO}(2-1)$	Davis et al. (2017a)

This paper has been typeset from a \LaTeX file prepared by the author.